

- Barcelona, Spain, September 3-7, 2018, Proceedings, Part II 23*, pp. 122–142, Springer, 2018.
- [25] J. D. P. Rodriguez and J. Posegga, “Rapid: Resource and api-based detection against in-browser miners,” in *Proceedings of the 34th Annual Computer Security Applications Conference*, pp. 313–326, 2018.
 - [26] A. Kharraz, Z. Ma, P. Murley, C. Lever, J. Mason, A. Miller, N. Borisov, M. Antonakakis, and M. Bailey, “Outguard: Detecting in-browser covert cryptocurrency mining in the wild,” in *The World Wide Web Conference*, pp. 840–852, 2019.
 - [27] D. Wang, B. Jiang, and W. Chan, “Wana: Symbolic execution of wasm bytecode for cross-platform smart contract vulnerability detection,” *arXiv preprint arXiv:2007.15510*, 2020.
 - [28] W. Chen, Z. Sun, H. Wang, X. Luo, H. Cai, and L. Wu, “Wasai: uncovering vulnerabilities in wasm smart contracts,” in *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*, pp. 703–715, 2022.
 - [29] S. Cao, N. He, Y. Guo, and H. Wang, “WASMixer: Binary Obfuscation for WebAssembly,” *arXiv e-prints*, p. arXiv:2308.03123, Aug. 2023.
 - [30] A. Romano, D. Lehmann, M. Pradel, and W. Wang, “Wobfuscator: Obfuscating javascript malware via opportunistic translation to webassembly,” in *2022 IEEE Symposium on Security and Privacy (SP)* (SP), (Los Alamitos, CA, USA), pp. 1101–1116, IEEE Computer Society, may 2022.
 - [31] A. Hilbig, D. Lehmann, and M. Pradel, “An empirical study of real-world webassembly binaries: Security, languages, use cases,” *Proceedings of the Web Conference 2021*, 2021.
 - [32] J. Cabrera-Arteaga, M. Monperrus, T. Toady, and B. Baudry, “Webassembly diversification for malware evasion,” *Computers & Security*, vol. 131, p. 103296, 2023.
 - [33] D. Chen and W3C group, “WebAssembly documentation: Security.” <https://webassembly.org/docs/security/>, 2020. Accessed: 18 June 2020.
 - [34] S. Narayan, C. Disselkoen, D. Moghimi, S. Cauligi, E. Johnson, Z. Gang, A. Vahldiek-Oberwagner, R. Sahita, H. Shacham, D. Tullsen, *et al.*, “Swivel: Hardening webassembly against spectre,” in *USENIX Security Symposium*, 2021.
 - [35] D. Genkin, L. Pachmanov, E. Tromer, and Y. Yarom, “Drive-by key-extraction cache attacks from portable code,” *IACR Cryptol. ePrint Arch.*, vol. 2018, p. 119, 2018.

- [36] Q. Stiévenart, C. De Roover, and M. Ghafari, “Security risks of porting c programs to webassembly,” in *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, SAC ’22, (New York, NY, USA), p. 1713–1722, Association for Computing Machinery, 2022.
- [37] J. Cabrera Arteaga, P. Laperdrix, M. Monperrus, and B. Baudry, “Multi-Variant Execution at the Edge,” *arXiv e-prints*, p. arXiv:2108.08125, Aug. 2021.
- [38] J. Cabrera-Arteaga, N. Fitzgerald, M. Monperrus, and B. Baudry, “WASM-MUTATE: Fast and Effective Binary Diversification for WebAssembly,” *arXiv e-prints*, p. arXiv:2309.07638, Sept. 2023.
- [39] M. Willsey, C. Nandi, Y. R. Wang, O. Flatt, Z. Tatlock, and P. Panchekha, “Egg: Fast and extensible equality saturation,” *Proc. ACM Program. Lang.*, vol. 5, jan 2021.
- [40] J. Cabrera Arteaga, “Artificial software diversification for webassembly,” 2022. QC 20220909.
- [41] M. Jacob, M. H. Jakubowski, P. Naldurg, C. W. N. Saw, and R. Venkatesan, “The superdiversifier: Peephole individualization for software protection,” in *International Workshop on Security*, pp. 100–120, Springer, 2008.
- [42] M. Henry, “Superoptimizer: a look at the smallest program,” *ACM SIGARCH Computer Architecture News*, vol. 15, pp. 122–126, Nov 1987.
- [43] R. Sasnauskas, Y. Chen, P. Collingbourne, J. Ketema, G. Lup, J. Taneja, and J. Regehr, “Souper: A Synthesizing Superoptimizer,” *arXiv preprint 1711.04422*, 2017.
- [44] E. Johnson, D. Thien, Y. Alhessi, S. Narayan, F. Brown, S. Lerner, T. McMullen, S. Savage, and D. Stefan, “Sfi safety for native-compiled wasm,” *NDSS. Internet Society*, 2021.
- [45] R. Sasnauskas, Y. Chen, P. Collingbourne, J. Ketema, G. Lup, J. Taneja, and J. Regehr, “Souper: A Synthesizing Superoptimizer,” *arXiv e-prints*, p. arXiv:1711.04422, Nov. 2017.
- [46] D. Cao, R. Kunkel, C. Nandi, M. Willsey, Z. Tatlock, and N. Polikarpova, “Babble: Learning better abstractions with e-graphs and anti-unification,” *Proc. ACM Program. Lang.*, vol. 7, jan 2023.
- [47] T. Rokicki, C. Maurice, M. Botvinnik, and Y. Oren, “Port contention goes portable: Port contention side channels in web browsers,” in *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS ’22, (New York, NY, USA), p. 1182–1194, Association for Computing Machinery, 2022.